



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
PEZZINI)
Serial No. 10/717,177)
Confirmation No. 5985)
Filing Date: November 19, 2003)
For: METHOD FOR GENERATING INTERRUPT)
COMMANDS IN A MICROPROCESSOR)
SYSTEM AND RELATIVE PRIORITY)
INTERRUPT CONTROLLER)


TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

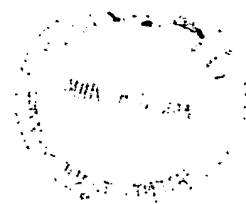
MS MISSING PARTS
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

Sir:

Transmitted herewith is a certified copy of the
priority European Application No. 02425709.9.

Respectfully submitted,


MICHAEL W. TAYLOR
Reg. No. 43,182
Allen, Dyer, Doppelt, Milbrath
& Gilchrist, P.A.
255 S. Orange Avenue, Suite 1401
Post Office Box 3791
Orlando, Florida 32802
Telephone: 407/841-2330
Fax: 407/841-2343
Attorney for Applicant



In re Patent Application of:

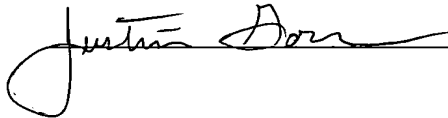
PEZZINI

Serial No. 10/717,177

Filing Date: November 19, 2003

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: MAIL STOP MISSING PARTS, COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450, on this 2nd day of March, 2004.





Europäisches
Patentamt

European
Patent Office

10.717.177
03-05-84
Office européen
des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

02425709.9

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk



Anmeldung Nr:
Application no.: 02425709.9
Demande no:

Anmeldetag:
Date of filing: 19.11.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

STMicroelectronics S.r.l.
Via C. Olivetti, 2
20041 Agrate Brianza (Milano)
ITALIE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se référer à la description.)

Method for generating interrupts in a microprocessor system and relative priority
interrupt controller

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s)
revendiquée(s)

Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06F9/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of
filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR

Italian Text Pursuant to Art. 14.2

“METODO DI GENERAZIONE DI SEGNALI DI INTERRUZIONE IN UN SISTEMA A MICROPROCESSORE E RELATIVO CONTROLLORE DI INTERRUZIONI CON PRIORITÀ”

5 CAMPO DELL'INVENZIONE

La presente invenzione concerne in generale i sistemi a microprocessore e più in particolare un metodo di generazione di segnali di interruzione in un sistema a microprocessore e un relativo controllore di interruzioni con priorità automaticamente incrementate in funzione del tempo di latenza e del tempo limite
10 entro cui la relativa interruzione deve essere servita.

BACKGROUND DELL'INVENZIONE

Durante il funzionamento di un processore, può essere necessario interrompere l'esecuzione del programma in corso per eseguire delle istruzioni particolari. Ciò viene fatto mediante dei segnali detti “interruzioni” (interrupt). Un controllore di
15 interruzioni riceve questi segnali e, in funzione dell'interruzione ricevuta, invia al processore un comando di interruzione e un vettore di interruzione che specifica l'indirizzo di memoria che contiene una relativa routine ISR (Interrupt Service Routine) da eseguire.

Il processore sospende l'operazione in corso, salva lo stato del programma che
20 stava eseguendo, in modo da poterlo riprendere in seguito, ed esegue le istruzioni contenute nella rispettiva routine ISR di servizio dell'interruzione, che varia a seconda dell'interruzione ricevuta. Quando la routine ISR è stata eseguita, il processore ripristina lo stato del programma e, se non ci sono altre interruzioni pendenti, riprende da dove era stato interrotto.

25 Comunemente, i controllori di interruzione hanno dei registri di priorità che permettono di stabilire quale interruzione, tra più interruzioni ricevute e pendenti, deve essere servita per prima.

Uno schema di principio di un noto controllore di interruzione con priorità è illustrato in Figura 1. Le interruzioni INT_0, \dots, INT_m provenienti dalle periferiche sono caricate in un registro delle interruzioni pendenti $INT\ PENDING\ REG$.

5 Il blocco $IRQ\ MASK\ AND\ PRIORITY\ LOGIC$ comprende sia la cosiddetta "maschera delle interruzioni", sia una logica di priorità che riceve un'interruzione insieme al suo livello di priorità, fornito dai registri di priorità $PRIORITY\ REGISTERS$. La logica di priorità genera un segnale di richiesta di interruzione $IRQ\ REQ$ e memorizza nel registro $CURR\ IRQ\ PRIORITY\ REG$ la relativa priorità $HIGHEST\ PRIORITY\ INT$.

10 Nel riquadro tratteggiato è rappresentato un circuito di elaborazione del segnale di richiesta di interruzione e della relativa priorità $HIGHEST\ PRIORITY\ INT$. Esso comprende una macchina a stati $IRQ\ SM$, che costituisce il cuore del controllore, che riceve il segnale di richiesta di interruzione ed invia un comando di interruzione $nIRQ$ al processore. Il segnale di richiesta d'interruzione $IRQ\ REQ$
15 seleziona un vettore di interruzione $IRQ\ VECTOR$ corrispondente all'interruzione richiesta, letto da una tabella delle interruzioni $IRQ\ VECTOR\ REG$ contenente vettori di interruzione identificanti routines ISR .

In Figura 1 sono rappresentati un registro $CURR\ IRQ\ PRIORITY\ REG$ e un
registro $PRIORITY\ STACK$ che servono essenzialmente per permettere le
20 cosiddette interruzioni nidificate ("nested interrupts"). Il registro $CURR\ IRQ\ PRIORITY\ REG$ memorizza la priorità dell'interruzione correntemente servita. Se viene generata un'interruzione con priorità più alta, il servizio della prima interruzione attualmente servita viene sospesa e la relativa priorità memorizzata nel registro $PRIORITY\ STACK$, mentre la nuova interruzione viene servita e la
25 relativa priorità memorizzata nel registro $CURR\ IRQ\ PRIORITY\ REG$.

Una volta servita la seconda interruzione, viene servita l'interruzione precedentemente sospesa, sempreché essa abbia priorità più alta rispetto alle interruzioni pendenti. Una volta che essa è stata servita, viene cancellata dallo stack $PRIORITY\ STACK$ mediante un comando $STACK\ PUSH/POP$ della

macchina a stati IRQ SM.

Un parametro importante dei controllori di interruzione è il tempo di latenza delle interruzioni, cioè il tempo trascorso da quando l'interruzione è stata ricevuta nel registro INT PENDING REG e l'istante in cui essa viene servita. Chiaramente, è
5 sempre desiderabile che questo tempo sia quanto più breve è possibile. Inoltre un'interruzione deve necessariamente essere servita entro un tempo massimo ("dead line") dal momento in cui viene caricata nel registro delle interruzioni pendenti, altrimenti l'applicazione governata dal programma in corso d'esecuzione potrebbe non funzionare correttamente.

10 Per evitare che un'interruzione venga eseguita dopo il tempo massimo prestabilito, in alcuni controllori noti i registri di priorità sono riprogrammati ad intervalli prestabiliti che in generale variano da registro a registro, aumentando il livello di priorità delle interruzioni all'aumentare del loro tempo di latenza nel registro delle interruzioni pendenti. In questo modo, le interruzioni da più tempo
15 in attesa hanno priorità più alta di quelle appena ricevute e sono servite prima che il loro tempo di latenza raggiunga il massimo consentito ("dead line").

Le priorità delle interruzioni memorizzate nello stack PRIORITY STACK invece non vengono incrementate, per evitare che un'interruzione prima sospesa a favore di una seconda interruzione possa successivamente sospendere il servizio della
20 seconda interruzione, e così via.

Questa tecnica di riprogrammazione però è poco conveniente perché il compito di riprogrammare i registri di priorità grava sul processore, che viene così rallentato.

SCOPO E SOMMARIO DELL'INVENZIONE

È stato trovato ed è l'oggetto della presente invenzione un metodo di generazione
25 di comandi di interruzione in sistemi a microprocessore che consente di superare gli inconvenienti delle tecniche note che prevedono la riprogrammazione dei registri di priorità da parte del processore.

Con il metodo dell'invenzione è possibile incrementare automaticamente i valori

di priorità delle interruzioni in attesa di essere servite, senza dover far gravare sul processore questa operazione. Questo risultato viene ottenuto utilizzando dei contatori destinati a contenere i valori di priorità effettivamente usati per determinare quale interruzione va servita per prima.

- 5 Non appena un'interruzione viene ricevuta, un relativo contatore contenente il rispettivo valore di priorità è incrementato ad intervalli di tempo prefissati mediante un segnale di incremento.

Il metodo dell'invenzione è implementato in un circuito di controllo di interruzioni con priorità adatto per un sistema a microprocessore, comprendente
10 registri di priorità memorizzanti valori di priorità associati a rispettive possibili interruzioni, uno o più registri delle interruzioni pendenti, una logica di priorità generante un segnale di richiesta di interruzione e un segnale interno rappresentativo della relativa priorità, e un circuito di elaborazione del segnale di interruzione e del segnale interno che invia ad un processore esterno un comando
15 di interruzione ed un vettore di interruzione.

Il circuito di controllo delle interruzioni dell'invenzione non necessita dell'intervento del processore per incrementare i valori di priorità di interruzioni pendenti perché comprende una pluralità di contatori accoppiati ai registri di priorità e inizializzati con i valori di priorità di interruzioni da servire. I contatori
20 ricevono ciascuno un segnale di incremento del loro contenuto e la logica di priorità legge dai contatori il livello di priorità associato a ciascuna interruzione.

I contatori sono preferibilmente incrementati da rispettivi segnali in quanto le priorità di ogni interruzione possono variare nel tempo in modo diverso da interruzione a interruzione.

- 25 L'invenzione è più precisamente definita nelle annesse rivendicazioni.

BREVE DESCRIZIONE DEI DISEGNI

I diversi aspetti e vantaggi dell'invenzione risulteranno ancor più evidenti attraverso una descrizione dettagliata dell'invenzione facendo riferimento ai

disegni allegati, in cui:

- la **Figura 1** è uno schema di un noto controllore di interruzioni;
- la **Figura 2** è uno schema generale di un controllore di interruzioni dell'invenzione;
- 5 la **Figura 3** illustra l'architettura interna di un controllore di interruzioni dell'invenzione.

DESCRIZIONE DI UNA FORMA DI REALIZZAZIONE DELL'INVENZIONE

Uno schema di principio del controllore di interruzioni INTERRUPT CONTROLLER dell'invenzione è mostrato in Figura 2. Diversamente dal
10 controllore noto di Figura 1, esso riceve in ingresso anche dei segnali PRIORITY TRIGGERS, che servono ad incrementare la priorità di interruzioni pendenti in modo da evitare che tali interruzioni restino pendenti per un tempo superiore ad un tempo massimo stabilito.

Una realizzazione preferita del controllore delle interruzioni dell'invenzione, è
15 mostrata in Figura 3. Esso si differenzia dal controllore noto di Figura 1 per la presenza di una pluralità di contatori di priorità PRIORITY COUNTERS, collegati ai registri di priorità e al circuito IRQ MASK AND PRIORITY LOGIC.

Quando nel registro INT PENDING REG è caricata un'interruzione, il contatore corrispondente, precedentemente caricato con il valore contenuto nel PRIORITY
20 REGISTER corrispondente, viene abilitato al conteggio. Il circuito IRQ MASK AND PRIORITY LOGIC legge dai contatori i valori di priorità, individua quale delle interruzioni pendenti ha la priorità più alta, genera un segnale di richiesta di interruzione IRQ REQ alla macchina a stati IRQ SM ed un segnale interno HIGHEST PRIORITY INT rappresentativo della priorità dell'interruzione INTn
25 che deve essere servita.

Allo stesso tempo, il contatore che memorizza il valore di priorità relativo all'interruzione INTn viene reinizializzato con il valore originario, il conteggio viene bloccato e l'interruzione INTn viene cancellata dal registro delle

interruzioni pendenti. Infine i valori di priorità delle interruzioni in attesa di essere servite, contenuti nei contatori PRIORITY COUNTERS, sono incrementati ad intervalli di tempo prestabiliti, che possono essere diversi da contatore a contatore, mediante i segnali di incremento PRIORITY TRIGGERS.

- 5 In questo modo la priorità di un'interruzione in attesa viene incrementata automaticamente via hardware al trascorrere del tempo. Così facendo, le interruzioni sono servite prima che il relativo tempo di latenza raggiunga il massimo consentito ("dead line"). Inoltre, diversamente dai controllori noti, con il controllore di interruzione dell'invenzione, il processore non è gravato dal
- 10 compito di riscrivere il contenuto dei registri di priorità.

Per permettere la gestione delle interruzioni nidificate ("nested interrupts"), il controllore preferibilmente avrà i registri CURR IRQ PRIORITY REG e PRIORITY STACK, come il controllore noto di Figura 1.

- 15 I segnali di incremento possono ad esempio essere derivati dal segnale di temporizzazione (clock) del controllore. Essi possono essere generati mediante divisori di frequenza oppure possono essere derivati dal segnale di temporizzazione proveniente dal sistema operativo, o segnali generati da eventi esterni o interni al microcontrollore (esempio il risultato di una comparazione o un segnale fornito su un pin).

- 20 Preferibilmente, ciascun contatore sarà incrementato da un rispettivo segnale di incremento, in modo da poter aumentare le priorità in modo diverso a seconda dell'interruzione a cui si riferiscono.

- Preferibilmente, il segnale di incremento di un contatore ha un periodo scelto in funzione del tempo massimo di latenza, in modo che la relativa interruzione possa
- 25 sicuramente raggiungere un valore massimo di priorità prima che sia trascorso il tempo limite predefinito. A tal scopo, questo periodo sarà convenientemente inferiore al rapporto tra il massimo tempo di latenza e la differenza tra un massimo ed un minimo valore di priorità previsti.

RIVENDICAZIONI

1. Metodo di controllo di interruzioni con priorità in un sistema a microprocessore, comprendente
associare preliminarmente valori di priorità prestabiliti a rispettive possibili
5 interruzioni (INT0, ..., INTm),
procurare un supporto di memorizzazione e memorizzare su di esso
interruzioni (INT0, ..., INTm) pervenute,
se almeno un'interruzione è memorizzata su detto supporto e se nessuna
routine di servizio di interruzione (ISR) è in corso d'esecuzione,
10 eseguire le seguenti operazioni:
confrontare tra loro i valori di priorità delle interruzioni memorizzate e
individuare quale di esse (INTn) ha il valore di priorità più alto,
generare un comando di interruzione (nIRQ) ed un vettore di interruzione
(IRQ VECTOR) identificante una rispettiva routine di servizio di
15 interruzione (ISR) da eseguire,
cancellare detta interruzione (INTn) con valore di priorità più alto da detto
supporto di memorizzazione,
eseguire detta routine di servizio di interruzione (ISR) identificata,
caratterizzato dal fatto che detto metodo comprende
20 procurare una pluralità di contatori;
per ogni nuova interruzione appena ricevuta da detto supporto, caricare in
un rispettivo contatore il corrispondente valore di priorità prestabilito;
fornire ad intervalli di tempo prestabiliti segnali di incremento ai contatori
memorizzanti valori di priorità di interruzioni pendenti in detto
25 supporto di memorizzazione;
detta operazione di individuazione dell'interruzione (INTn) con valore di
priorità più alto è eseguita confrontando i valori di priorità contenuti nei
contatori associati alle interruzioni memorizzate.
2. Il metodo della rivendicazione 1, in cui detti segnali di incremento sono
30 forniti ad intervalli di tempo uguali la cui durata è determinata in funzione di un

tempo massimo di latenza delle interruzioni in detto supporto di memorizzazione.

3. Il metodo della rivendicazione 2, in cui la durata di detti intervalli di tempo è inferiore al rapporto tra detto tempo massimo di latenza e la differenza tra un massimo ed un minimo valore di priorità.

5 4. Il metodo di della rivendicazione 1 di interruzioni nidificate, comprendente inoltre

 procurare un buffer di memoria (CURR IRQ PRIORITY REG);

 procurare un registro stack (INT PRIORITY STACK);

 memorizzare in detto buffer di memoria (CURR IRQ PRIORITY REG) la
10 priorità di un'interruzione che si sta servendo;

 detta operazione di individuazione dell'interruzione (INT_n) con valore di
 priorità più alto è eseguita confrontando i valori di priorità contenuti nei
 contatori associati alle interruzioni memorizzate e in detto buffer di
 memoria (CURR IRQ PRIORITY REG);

15 qualora un'interruzione dovesse avere una priorità più alta dell'interruzione
 che si sta servendo, interrompere il servizio dell'interruzione in corso e
 memorizzare in detto registro stack (INT PRIORITY STACK) la
 priorità dell'interruzione il cui servizio è stato sospeso;

 eseguire l'interruzione corrispondente alla priorità in testa al registro stack
20 non appena detta priorità è la più alta tra le priorità delle interruzioni
 pendenti, e cancellare da detto registro stack (INT PRIORITY STACK)
 la priorità dell'interruzione servita.

5. Circuito di controllo di interruzioni con priorità in un sistema a
microprocessore, comprendente

25 registri di priorità (PRIORITY REGISTERS) memorizzanti valori di
 priorità associati a rispettive possibili interruzioni (INT₀, ..., INT_m),
 almeno un registro delle interruzioni pendenti (INT PENDING REG),
 memorizzante interruzioni (INT₀, ..., INT_m) da parte di periferiche
 collegate al circuito di controllo,

30 una logica di priorità (IRQ MASK AND PRIORITY LOGIC) accoppiata a

- 5 detti registri di priorità (PRIORITY REGISTERS) e a detto registro delle interruzioni pendenti (INT PENDING REG), generante un segnale di richiesta di interruzione (IRQ REQ) e un segnale interno (HIGHEST PRIORITY INT) rappresentativo della priorità dell'interruzione (INT_n) memorizzata in detto registro delle interruzioni pendenti (INT PENDING REG) avente il livello di priorità più alto,
- 10 un circuito di elaborazione di detto segnale di richiesta di interruzione (IRQ REQ) e di detto segnale interno (HIGHEST PRIORITY INT), inviante ad un processore esterno un comando di interruzione (nIRQ) ed un vettore di interruzione (IRQ VECTOR) identificante una rispettiva routine di servizio di interruzione (ISR) da eseguire, caratterizzato dal fatto che comprende inoltre
- 15 una pluralità di contatori (PRIORITY COUNTERS) accoppiati a detti registri di priorità (PRIORITY REGISTERS) e inizializzati con i valori di priorità di interruzioni (INT₀, ..., INT_m) da servire, detti contatori (PRIORITY COUNTERS) ricevendo segnali di incremento (PRIORITY TRIGGERS) del loro contenuto;
- 20 detta logica di priorità (IRQ MASK AND PRIORITY LOGIC) leggendo da detti contatori (PRIORITY COUNTERS) il livello di priorità associato a ciascuna interruzione (INT₀, ..., INT_m) memorizzata in detto registro delle interruzioni pendenti (INT PENDING REG).
- 25 6. Il circuito di controllo della rivendicazione 5, in cui detti segnali di incremento (PRIORITY TRIGGERS) comprendono un segnale di temporizzazione del controllore e segnali esternamente generati forniti su pin d'ingresso di detto circuito di controllo.
7. Il circuito di controllo della rivendicazione 5, in cui detti segnali di incremento (PRIORITY TRIGGERS) hanno periodo multiplo di quello di un segnale di temporizzazione del controllore.
- 30 8. Il circuito di controllo della rivendicazione 5, in cui ciascuno di detti

contatori (PRIORITY COUNTERS) riceve un rispettivo segnale d'incremento.

9. Il circuito di controllo della rivendicazione 5 di interruzioni nidificate, in cui detto circuito di elaborazione comprende

5 un buffer di memoria (CURR IRQ PRIORITY REG) accoppiato a detta logica di priorità (IRQ MASK AND PRIORITY LOGIC), in cui memorizzare la priorità di un'interruzione da servire rappresentata da detto segnale interno (HIGHEST PRIORITY INT);

10 un registro stack (INT PRIORITY STACK) accoppiato a detto buffer di memoria (CURR IRQ PRIORITY REG) memorizzante la priorità di interruzioni il cui servizio è stato sospeso da un'interruzione con priorità più alta;

15 una macchina a stati (IRQ SM) ricevente detto segnale di richiesta di interruzione (IRQ REQ), generante detto comando di interruzione (nIRQ) e un comando di controllo (STACK PUSH/POP) di detto registro stack (INT PRIORITY STACK).

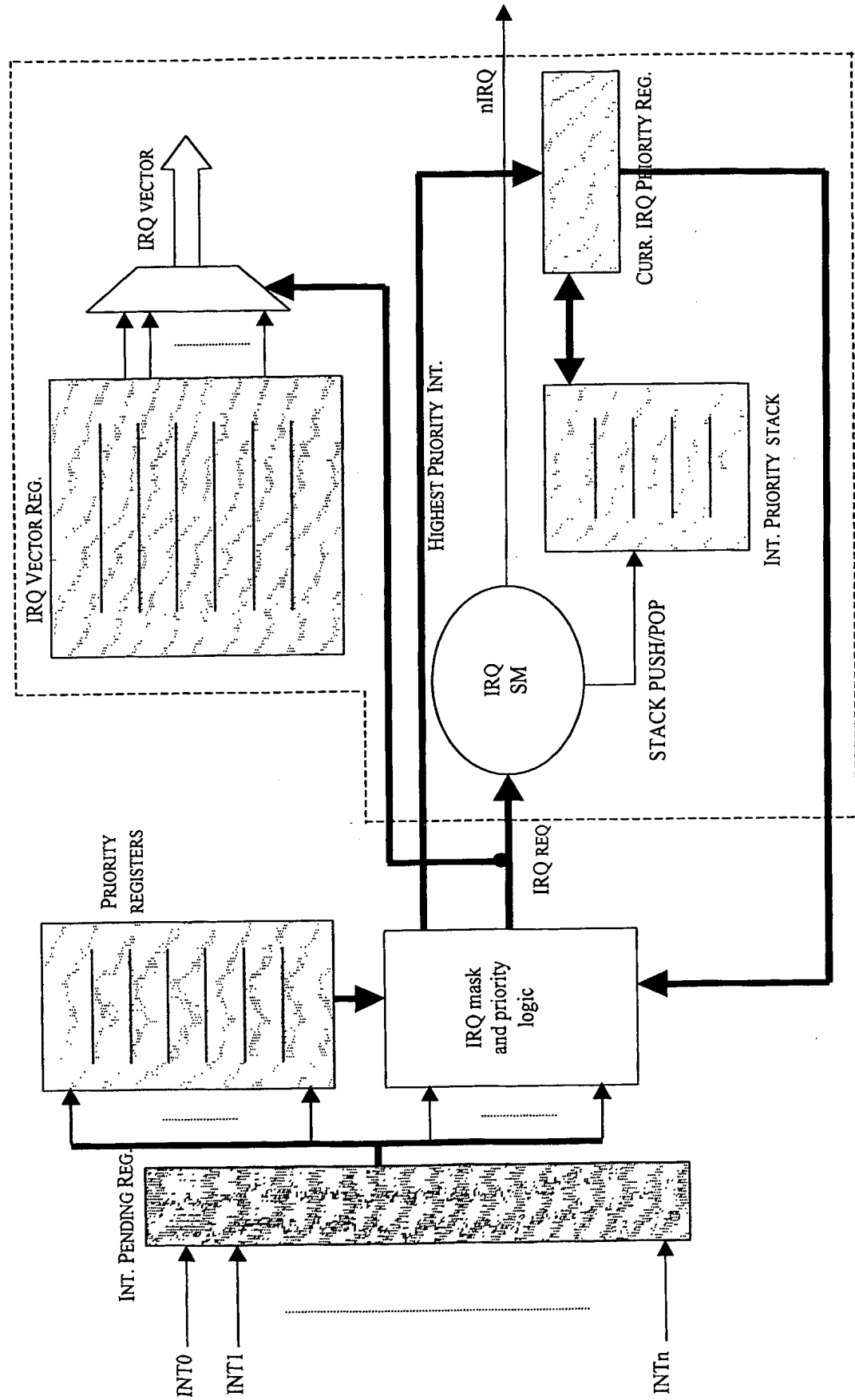
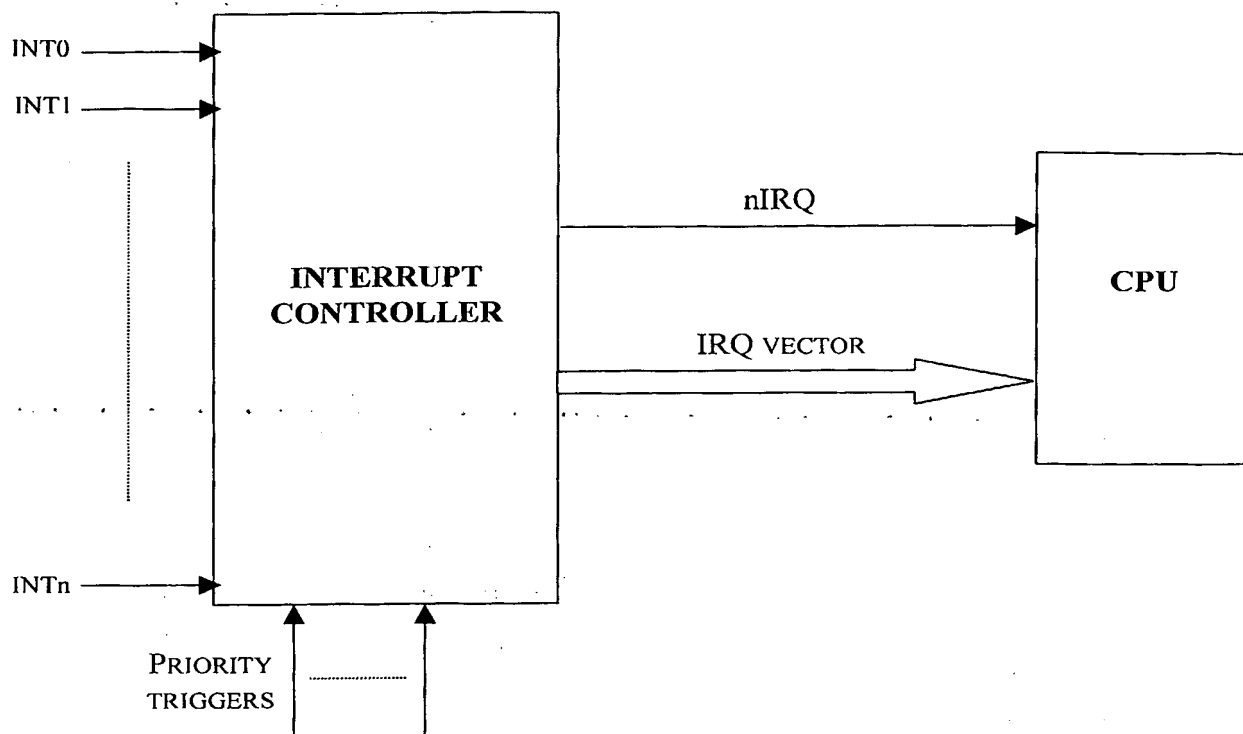


FIG. 1

**FIG. 2**

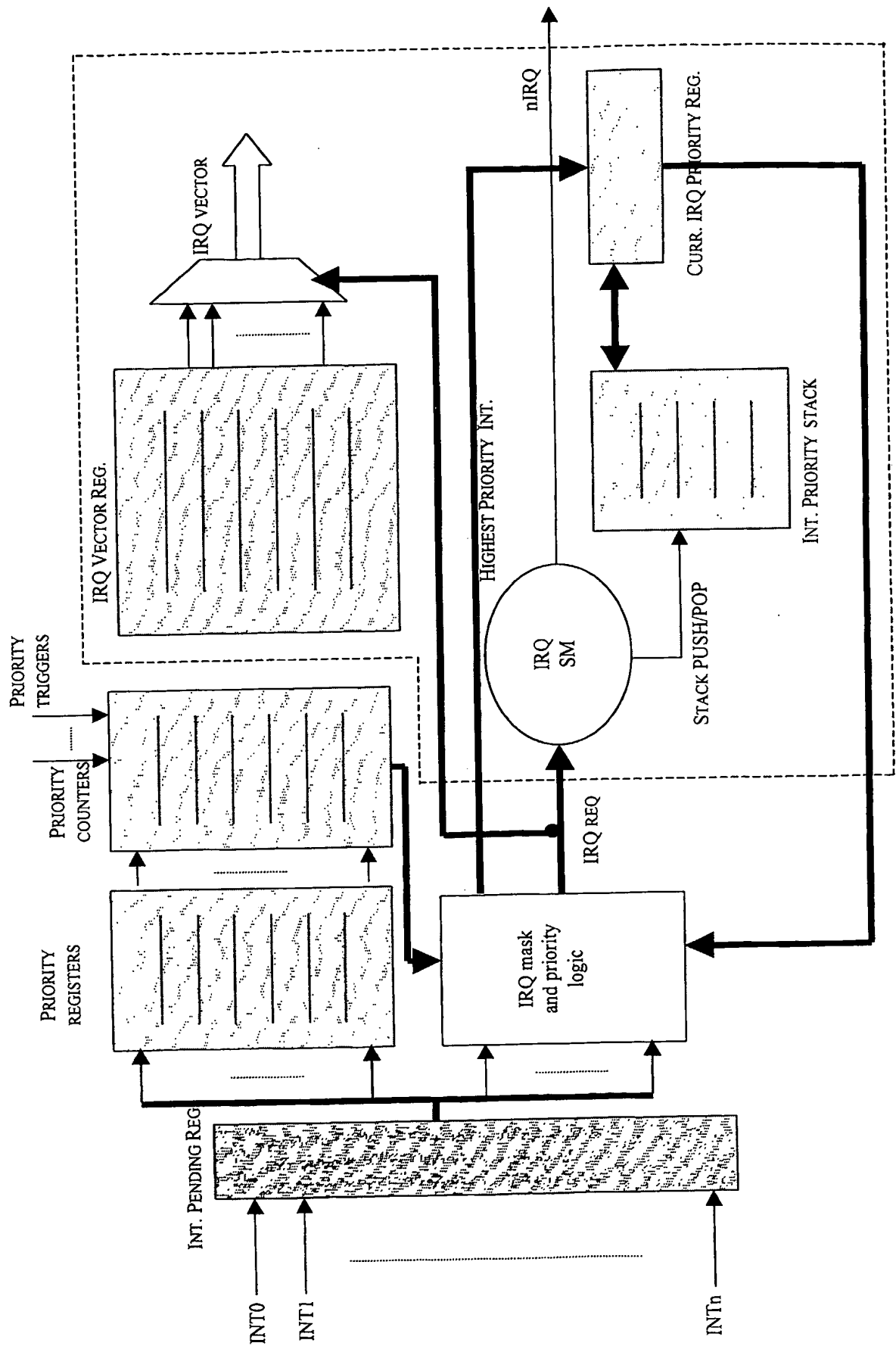


FIG. 3

